CMPE353/CMSE354

RELATIONAL DATABASE DESIGN EXAMPLES
(Ch. 8)

Problem at hand

We will consider the following set F of functional dependencies for relation schema $R = (A, B, C, D, E)$ throughout these examples.

$F=\{A \rightarrow BC, CD \rightarrow E, B \rightarrow D, E \rightarrow A\}$

Ex. 1 Compute B^+

- Using the attribute closure algorithm below, we find that B^+=BD

F={A → BC, CD → E, B → D, E → A}

```
result := α;
        while (changes to result) do{
            for each β → γ in F do
              begin
                if β ⊆ result then

              result := result ∪ γ
              end
               }
α+=result
```

result=B
result=BD

B^+=BD

Remember! This means that the functional dependencies B→B, B→D and B→BD are all in F^+

Ex.2 Is BC a candidate key for R?

- Using the attribute closure algorithm we find that $(BC)^+=ABCDE$

  Since the closure contains all attributes in R, BC is a superkey.
- Now, we test if any subset of BC, that is B or C is a superkey for R?

  If not, BC will be a candidate key.

  - We already know from Ex.1 that $B^+=BD$, so B is not a SK.
  - Using attribute closure algoritm, we find that $C^+=C$, so C is not a SK.

  - Since no subset of BC is SK, BC becomes the candidate key!

Additional information.

- We can show that *A, BC, CD,* and *E* are all candidate keys for R.

- Try this!

<u>Ex. 3</u> Let R1 = (A, B, C), R2 = (A, D, E). Is this decomposition of R lossless join decomposition?

- We know that a decomposition *{R1, R2}* is a lossless-join decomposition if $R1 \cap R2 \rightarrow R1$ or $R1 \cap R2 \rightarrow R2$.

(In other words if $R1 \cap R2$ is a superkey for R1 or R2)

- $R1 \cap R2$=A
- We know from slide 5, that A is superkey for R, so it must be a superkey for any decomposition of R. (R1 and R2 in this case)
- So the decomposition is lossless join.
- <u>Alternatively</u>;
  - Use the only functional dependency(FD) in F defined over R1: $A \rightarrow BC$
  - Compute A^+=ABC. Since all attributes of R1 are in A^+, A is a SK for R1.

Ex.4 Is R= (A, B, C, D, E) in BCNF given F={A → BC, CD → E, B →D, E →A}?

A relation schema *R* is in BCNF with respect to a set *F* of functional dependencies if for all functional dependencies in *F* of the form

$$\alpha \rightarrow \beta$$

where $\alpha \subseteq R$ and $\beta \subseteq R$, at least one of the following holds:

1) $\alpha \rightarrow \beta$ is trivial (i.e., $\beta \subseteq \alpha$)

OR

2) α is a superkey for *R*

Ex. 4 Solution Continued

- F={A → BC, CD → E, B →D, E →A}
  We must check each FD to see if iy satisfies cond. 1 or 2.
- If they all satisfy, then R is BCNF.
  - Checking A → BC; we know that A is a SK for R (slide 5); 2 is satisfied.
  - Checking CD → E; we know that CD is a SK for R (slide 5); 2 is satisfied.
  - Checking B →D; B is not SK and is non-trivial. Both 1 and 2 are not satisfied.
  - Checking E → A; we know that E is a SK for R (slide 5); 2 is satisfied.

Since we found at least one FD in F which doesn't satisfy any of the conditions, we conclude that R is not BCNF.

Ex.5 Use BCNF decomposition algorithm once to decompose R in Ex.4 into R1 and R2.

- We decompose *R* into: assume ($\alpha \rightarrow \beta$ *)*

  R1=(α U β), R2=(*R* - (β - α)) using the functional dependency which violates BCNF.

  - In Ex.4 B →D was the violating FD, so;

  - R1=(α U β) =(B U D)=(B,D)
  - R2 =(R - (β - α))= (ABCDE)-(D-B)=(A,B,C,E)

  - *(Do not check if this decomposition is BCNF since this requires all FD in the closure of F, which we do not know!)*

Ex.6 Is R = (A, B, C, D, E) in 3NF given F={A → BC, CD → E, B →D, E →A} ?

- A relation schema *R* is in **third normal form (3NF)** if for all:

  $\alpha \rightarrow \beta$ in F^+

  at least one of the following holds:

  1) $\alpha \rightarrow \beta$ is trivial (i.e., $\beta \in \alpha$)
  2) α is a superkey for *R*
  3) Each attribute *A* in $\beta - \alpha$ is contained in a candidate key for *R*.

     (**NOTE**: each attribute may be in a different candidate key)

Ex. 6 solution continued

- We know from Ex.4 that FDs A → BC, CD → E, E →A all satisfy condition 2. But the FD B →D doesn't satisfy 1 or 2. We check if it satisfies condition 3.

3) Each attribute *A* in $\beta - \alpha$ is contained in a candidate key for *R*

$\beta - \alpha$ =D-B=D

So we check if attribute D is in any candidate key for R. We know that CD is a candidate key for R (slide 5). Therefore D is contained in candidate key CD. So condition 3 is satisfied.

- We conclude that R is 3NF.
- Nevertheless we can still give a 3NF decomposition of R (Ex.7 next)

Ex.7 Give a 3NF decomposition of R=(A,B,C,D,E) given F={A → BC, CD → E, B →D, E →A}

- A simplified version of the 3NF decomposition may be written in the following way:
- Step 1) For each FD $\alpha \rightarrow \beta$ *in F* create a Ri=(α,β)
    - Using every FD in F we get: R1=(ABC), R2=(CDE), R3=(BD), R4=(AE)
- Step 2) If none of the previously formed schemas contains a candidate key for R, form an additional Rj to include one of the candidate keys.
    - Since several candidate keys (not just one) are included in R1, R2,R3, R4 we do not need to form an additional R5.
- Step 3) If any schema R_j is contained in another schema R_k previosly formed, remove schema R_j
    - Here we have no such schema, so we do not remove any.
- We conclude that,R1=(ABC), R2=(CDE), R3=(BD), R4=(AE) is a 3NF decomposition of R.
-

Ex.8 Given F={A → BC, CD → E, B → D, E → A} determine i) if attribute C in CD → E is extraneous ii) if attribute B in A → BC is extraneous

- We use the following tests to check for extraneous attributes:
- **Left hand side attribute test:**
- To test if attribute $A \in \alpha$ is extraneous in α compute $(\{\alpha\} - A)^+$ using the dependencies in *F*
  1. check that $(\{\alpha\} - A)^+$ contains β; if it does, *A* is extraneous in α
- **Right hand side attribute test:**
- To test if attribute $A \in \beta$ is extraneous in β
  1. compute α^+ using only the dependencies in
     $F' = (F - \{\alpha \rightarrow \beta\}) \cup \{\alpha \rightarrow (\beta - A)\}$,
  2. check that α^+ contains *A*; if it does, *A* is extraneous in β

Ex. 8 solution continued

i) is attribute C in CD → E is extraneous ?

- Apply LHS and compute CD-C=D^+ using F by attribute closure algorithm.
- D^+=D.
- Since it doesn't contain all attributes on the RHS (i.e. attribute E), we conclude that C is not extraneous.

Ex. 8 solution continued

ii) is attribute B in A → BC is extraneous?

- Apply RHS and compute A^+ using F' by attribute closure algorithm.
- F'= {A → C, CD → E, B →D, E →A}
- A → BC is replaced by A → C (i.e. Remove the attribute under test from the FD it belongs to; leave all others unchanged.
- A^+=AC
- Since it doesn't contain attribute under test (i.e attribute B) we conclude that B is not extraneous.